<u>CLAIM AMENDMENTS</u>

This listing of claims will replace all prior versions, and
listings, of claims in the application:

<u>Listing of Claims:</u>

Claim 1 (currently amended). A microprocessor circuit,
comprising:

at least one control unit;

at least one memory for free programming with at least one
program having functions, said memory connected to said
control unit;

a stack for buffer storing data, said stack connected to said
control unit;

a register bank having registers, said register bank connected
to said control unit; and

an auxiliary register storing a number of bits, each of the
bits being associated with one of said registers of said
register bank and indicating whether a respective one of said
registers contains a value different from a logical "0", said

auxiliary register connected to at least one of said control

unit, or said register bank, and ~~said stack~~

a stack buffer-storing data of at least one of said auxiliary

register or said registers of said register bank, said stack

connected to at least one of said control unit, said register

bank, or said auxiliary register.

Claim 2 (original). The microprocessor circuit according to

claim 1, wherein said auxiliary register has a number of

further registers corresponding to a number of said registers

of said register bank.

Claim 3 (original). The microprocessor circuit according to

claim 2, wherein each of said further registers store a single

bit.

Claim 4 (original). The microprocessor circuit according to

claim 1, wherein said auxiliary register has only one further

register for storing a bit sequence corresponding to a number

of said registers of said register bank.

Claim 5 (original). The microprocessor circuit according to

claim 4, wherein said auxiliary register is at least one of

said registers of said register bank.

6. (currently amended). The microprocessor circuit according to claim 1, further comprising a second stack for storing at least some data in said register bank, said second stack inaccessible by a programmer being accessible only to an operating system of said control unit.

Claim 7 (original). The microprocessor circuit according to claim 1, further comprising a second stack for storing at least some data in said register bank, said second stack inaccessible by a programmer.

Claim 8 (original). The microprocessor circuit according to claim 1, further comprising a second stack for storing at least some data in said register bank, said second stack connected to said control unit and inaccessible by the program.

Claim 9 (currently amended). The microprocessor circuit according to claim 1, wherein:

said register bank has first and second areas with first and second registers; and

said first registers storing data of a called function;

Application No. 10/622,981
Amdt. dated 12/14/05
Reply to Office action of 10/14/05

said second registers storing data of both a called function

and data of a calling function; and

at least said first registers are associated with a bit in

said auxiliary register.

Claim 10 (currently amended). The microprocessor circuit

according to claim 1, wherein:

said register bank has first and second areas with first and

second registers; ~~and~~

said first registers storing data of a called function;

said second registers storing data of both a called function

and data of a calling function; and

at least said first registers are associated with said

auxiliary register.

Claim 11 (original). The microprocessor circuit according to

claim 9, wherein said first area of said register bank is

divided into a plurality of sub-areas respectively available

to one of the functions of the program.

Page 5 of 16

Application No. 10/522,981
Amdt. dated 12/14/05
Reply to Office action of 10/14/05

Claim 12 (currently amended). A method for operating a
microprocessor circuit, which comprises the steps of:

providing a circuit with:

   at least one control unit;

   at least one memory for free programming with at least
   one program having functions;

   a stack for buffer-storing data;

   a register bank having registers; and

   an auxiliary register;

setting all of the bits of the auxiliary register to a logical
"0" when the circuit is initialized;

storing bits in the auxiliary register, each of the bits being
associated with one of the registers and indicating whether a
respective one of the registers contains a value different
from a logical "0"; and

Page 6 of 16

setting a bit of the auxiliary register associated with a

respective one of the registers to a value different from the

logical "0" when a datum is written to the associated register

of the register bank; and

storing data being stored in at least one of the auxiliary

register or the registers of the register bank in the stack,

or storing data which is stored in the stack in at least one

of the auxiliary register or the registers of the register

bank.

13. (original) The method according to claim 12, which further

comprises permitting a read of a datum from one of the

registers of the register bank only if the associated bit of

the auxiliary register has the value different from the

logical "0".

14. (original) The method according to claim 12, which

further comprises, when a datum from one of the registers of

the register bank whose associated bit of the auxiliary

register has the logical "0" value is read, returning the

datum "0".

15. (original) The method according to claim 13, which

further comprises, when a datum from one of the registers of

the register bank whose associated bit of the auxiliary

register has the logical "0" value is read, returning the

datum "0".

16. (original) The method according to claim 12, which

further comprises:

providing the circuit with a second stack for storing at least

some data in the register bank and making the second stack

inaccessible by a programmer; and

if the circuit changes from a first function to a second

function, successively storing the data associated with the

first function in the registers of the register bank and the

bit sequence of the auxiliary register in one of the stack and

the second stack.

17. (original) The method according to claim 16, which

further comprises setting the bits of the auxiliary register

to the logical "0" value after storing the data and the bit

sequence of the auxiliary register in one of the stacks.

18. (new) The method according to claim 12, wherein a

register of the register bank is only stored on the stack if

the associated bit of the auxiliary register has a value

different from the logical "0".

19. (new) The method according to claim 18, wherein the auxiliary register is stored as the last value on the stack.

20. (new) The method according to claim 14, wherein a register of the register bank is only stored on the stack if the associated bit of the auxiliary register has the value different from the logical "0".

21. (new) The method according to claim 14, wherein the auxiliary register is stored as the last value on the stack.